International Journal of Safety and Security Engineering
Vol. 15, No. 2, February, 2025, pp. 369-382
Journal homepage: http://iieta.org/journals/ijsse

Design and Develop the Business Process Model for Open Access Copyright Management System Using Permission Less Blockchain

K. Varaprasada Rao[1*], Dileep Kumar Murala[1], Sandeep Kumar Panda[2]

[1] Department of Computer Science and Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad 501203, India
[2] Department of Artificial Intelligence and Data Science, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education, Hyderabad 501203, India

Corresponding Author Email: varaprasad.fst@gmail.com

https://doi.org/10.18280/ijsse.150218

Received: 26 December 2024
Revised: 26 January 2025
Accepted: 10 February 2025
Available online: 28 February 2025

Keywords:
Blockchain Technology (BT), smart contacts, Copyright Management System (CMS), Business Process Model (BPM), security, decentralization, transparent

ABSTRACT

Copyright management is essential in the digital economy for protecting intellectual property rights. With the exponential growth of digital content, existing centralized systems struggle with inefficiencies, enforcement challenges, and a lack of transparency. Blockchain technology offers a decentralized and tamper-proof solution, enabling transparent, traceable, and legally compliant copyright management. However, achieving a standardized, trusted, and interoperable platform remains a key challenge. This study presents an optimized blockchain-based copyright management model that enhances process coordination and automation. The approach utilizes a structured "Chain of Transformation", converting an optimized state transition model into intelligent smart contracts. Key stages include: (1) defining the business process state transition model, (2) optimizing it using the Processes States and Transition Reduction Algorithm (PSTRA), (3) transforming it into smart contracts, and (4) refining these contracts into intelligent contracts. This structured automation ensures secure, efficient, and enforceable copyright management. Comparative analysis with existing systems highlights improvements in decentralization, security, cost efficiency, process communication, and standardization. The integration of intelligent smart contracts enhances automation, ensuring transparent and legally enforceable copyright execution. The model also strengthens security against copyright infringements by leveraging blockchain's immutable ledger for verification and enforcement. The proposed blockchain-based model improves copyright protection by providing a secure, automated, and legally compliant framework. It fosters innovation by standardizing copyright processes, reducing inefficiencies, and lowering operational costs. Future research will focus on enhancing scalability, cross-chain interoperability, and legal adaptability to address evolving industry needs.

1. INTRODUCTION

A Copyright Management System (CMS) ensures secure handling of intellectual property rights, vital for protecting creators' work. Integrating a CMS with a business process model using blockchain technology represents a transformative approach to managing digital rights in a decentralized, secure, and efficient manner [1]. This innovative integration leverages the strengths of blockchain, such as immutability, transparency, and distributed ledger technology, to address longstanding challenges in copyright management, including unauthorized use, data tampering, and inefficient manual processes [2]. By converting business process models into smart contracts, this approach automates the enforcement and execution of copyright rules, ensuring consistent and error-free operations. Additionally, the decentralized nature of blockchain eliminates the risk of single points of failure, enhancing the overall resilience and reliability of the system [3]. This method not only streamlines operations and reduces costs by removing intermediaries but also significantly boosts trust and accountability among stakeholders through transparent and verifiable transactions. As a result, integrating blockchain technology into copyright management and business processes offers a comprehensive solution that enhances security, efficiency, and operational effectiveness in the digital age [4].

1.1 Copyright management system

A Copyright management system (CMS) is designed to oversee the administration, protection, and monetization of copyrighted content, including music, literature, software, and digital media [5]. It enables key functions such as copyright registration, rights management, licensing, and royalty tracking to ensure fair compensation for creators. Core components include a secure database for precise record-keeping, a user-friendly interface for seamless access, and automated workflows to streamline tasks like license allocation and royalty computation. Advanced technologies such as Digital Rights Management (DRM), blockchain for secure transactions, and AI for content recognition enhance efficiency and security [6]. Cloud computing further supports scalability and processing capabilities.

Effective CMS implementation requires compliance with copyright laws, an intuitive interface, scalable infrastructure, interoperability with other systems, and strong data privacy measures [7]. By integrating these elements, a CMS enhances copyright protection, streamlines rights management, and optimizes content monetization. This benefits creators, rights holders, and consumers by fostering a transparent and efficient digital content ecosystem [8].

Figure 1 plays a vital role in protecting and managing intellectual property by integrating key functions essential for rights administration and enforcement [9]. Registration and documentation ensure a structured database that records crucial details such as titles, authors, and publishers, providing verifiable proof of ownership and copyright status. Rights management defines usage conditions, access controls, and permissions, regulating how copyrighted content can be utilized [10]. Licensing management facilitates granting and monitoring licenses, ensuring compliance with terms while securing fair compensation for rights holders. Royalty tracking and distribution automates usage monitoring, calculates royalties, and ensures accurate payments to copyright owners [11].

Additionally, compliance and enforcement mechanisms detect unauthorized usage, issue takedown notices, and support legal action to uphold copyright laws. Metadata management organizes essential details like creation dates and usage rights, improving searchability and tracking [12]. Figure 2 represents the Content identification employs technologies such as watermarking, fingerprinting, and blockchain to verify authenticity and ownership, ensuring secure copyright protection. These functions collectively strengthen intellectual property management, offering a transparent and efficient framework for content creators and rights holders.

A CMS is particularly important in industries where copyright is a significant concern, such as music, film, publishing, and software.

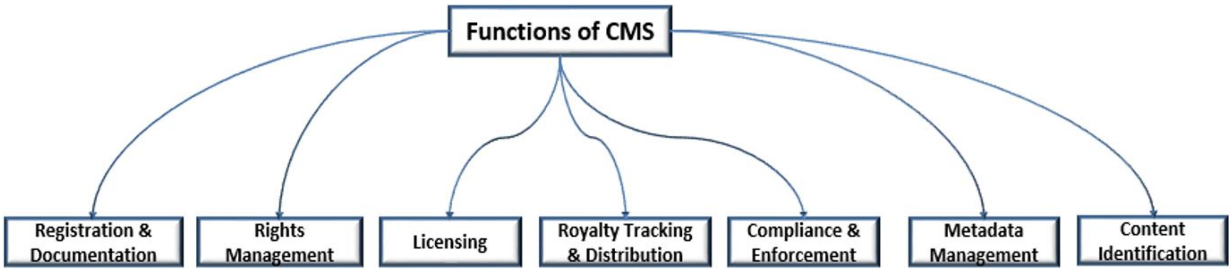

Figure 1. Function of CMS

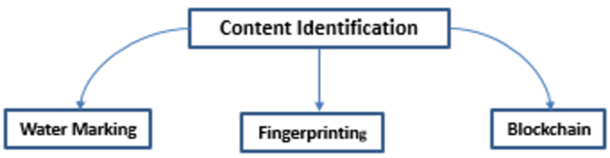

Figure 2. Content identification implementation

1.1.1 Issues and challenges of CMS

A CMS faces several challenges that impact its efficiency and effectiveness. Centralized control and dependency create bottlenecks, leading to lengthy registration processing times [13]. This makes it difficult for creators and rights holders to protect and manage their works promptly. Additionally, tracking and traceability of ownership are often complex and cumbersome, further complicating the management process. Disputes related to counterfeits and infringements are on the rise each year, highlighting the system's struggle to enforce rights adequately. Moreover, the methods for conducting copyright transactions and protecting rights are not optimized, resulting in inefficiencies and gaps in enforcement. These challenges underscore the need for improvements in copyright management practices to better serve creators, rights holders, and users [14].

1.2 Business process model

Business Process Management (BPM) is a methodology for continuously discovering, modelling, analysing, and optimizing system processes to enhance cooperation, collaboration, and efficiency [15]. A well-structured Business Process Model (BPM) improves operational performance by identifying and eliminating bottlenecks, redundancies, and inefficiencies, resulting in streamlined workflows. It ensures process standardization, aligning operations with industry best practices and fostering clear communication among stakeholders [16].

Additionally, BPM enables automation and system integration, facilitating seamless data flow and reducing manual intervention. It enhances risk management by identifying potential issues and implementing proactive mitigation strategies. Effective BPM also supports asset optimization, ensuring resources are utilized efficiently while minimizing waste. By improving efficiency, standardization, and risk management, BPM creates a more structured and high-performing system [17].

Figure 3 illustrates some of the benefits of the Business process model significantly enhances the overall effectiveness and efficiency of the system.

1.2.1 Integrating of business process model for CMS

By clearly defining processes, it helps streamline operations, ensuring smoother workflows and reducing redundancies [18]. This improvement leads to better protection, transparency, and security within the system, as processes are more visible and manageable. Additionally, the model supports scalability and performance, allowing the system to adapt and grow with the business's needs [19]. It also aids in revenue generation by optimizing processes, reducing costs, and improving service delivery. Interoperability is another key benefit, as the model facilitates integration with other systems, ensuring seamless data exchange and collaboration across different platforms [20]. Overall, the Business Process Model provides a comprehensive framework that enhances operational efficiency, security, and profitability. Figure 4 illustrates a multilayered architecture of Business process management system comprising the access layer, business process management layer and storage layer.

1.2.2 Architecture of business process model for CMS.

The Access Layer serves as the primary user interface, enabling users to initiate, monitor, and manage business processes [21]. Below it, the Business Process Management Layer oversees multiple workflows, such as Business Process 1, Business Process 2, and others, ensuring seamless execution and interaction with the Storage Layer for data retrieval and storage [22]. The Storage Layer comprises critical systems like MIS (Management Information Systems), HRMS (Human Resource Management Systems), CRM (Customer Relationship Management), and SAP (Systems, Applications, and Products in Data Processing), forming the backbone of business data infrastructure. This layered architecture facilitates structured data flow, improving operational efficiency and management.

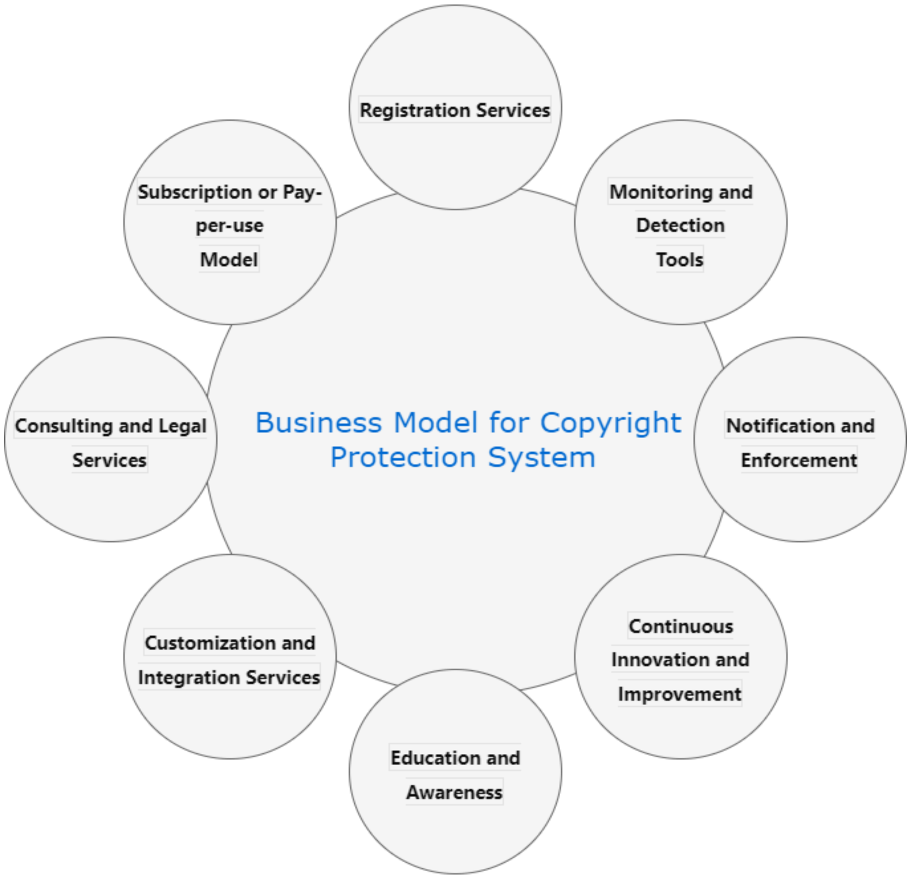

Figure 3. Business model of CMS functionalities

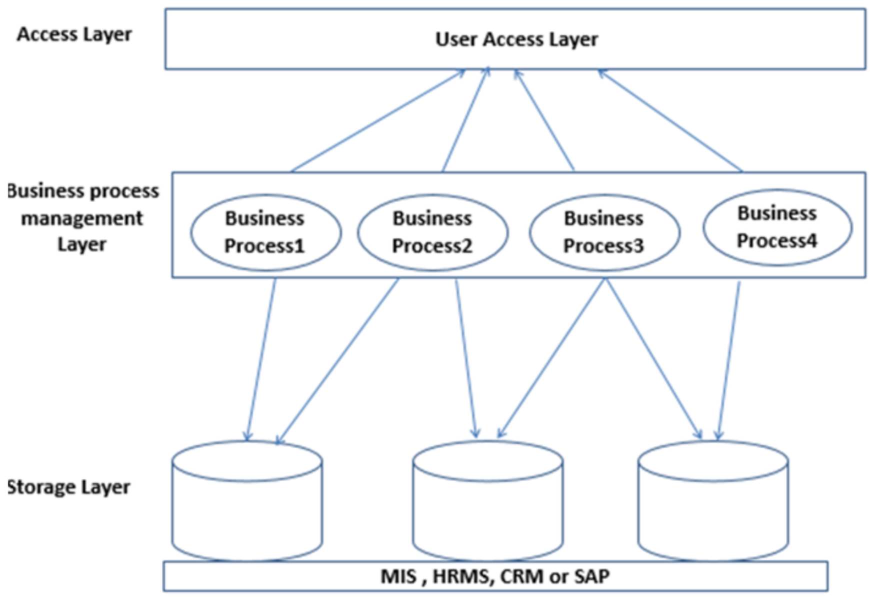

Figure 4. The architecture of business process model for CMS

However, several challenges persist, particularly in transparency and traceability within intra- and inter-organizational processes. The lack of end-to-end security exposes systems to breaches and unauthorized access. Additionally, inconsistent transaction recording creates accountability gaps, limiting oversight and process integrity [23]. Interoperability remains a major concern, as seamless integration across platforms is essential for efficient data exchange and system functionality. Addressing these issues will enhance the security, reliability, and overall efficiency of business process management.

1.3 Blockchain technology

Blockchain is a decentralized and distributed technology that enables secure digital asset transactions without intermediaries, ensuring transparency and trust among participants [24]. It achieves security through **consensus** (collective transaction validation), **provenance** (ownership tracking), **immutability** (tamper-proof records), and **finality** (synchronized, unalterable ledgers) [25]. Introduced by Satoshi Nakamoto in 2008, Bitcoin's blockchain revolutionized digital transactions by enabling decentralized mining and verification [26]. By 2010, Bitcoin gained real-world value when 10,000 BTC were exchanged for two pizzas [27]. Since then, alternative cryptocurrencies have enhanced transaction speed, anonymity, and decentralization, expanding blockchain's adoption across industries.

In Figure 5, blockchain networks rely on peer nodes to maintain the public ledger and execute smart contracts, which automate agreements using predefined conditions, typically written in solidity [28]. Transactions are validated through Proof-of-Work (PoW) or Proof-of-Stake (PoS) consensus mechanisms, ensuring a trustless system [29]. Various open-source platforms, including Ethereum, Hyperledger, and Quorum, support industry-specific applications. Miners use tools like Ethminer, OpenCL Mining, and Geth, while Web3.js enables Ethereum-based decentralized applications. Peer nodes access blockchain networks through Metamask and Mist, ensuring seamless interaction and security [30].

1.4 Research problems and objectives

Existing CMSs face several challenges, including the lack of a standardized framework, leading to inconsistencies in digital rights protection. Traditional systems are prone to security risks, unauthorized alterations, and lack transparency, making ownership verification difficult. Additionally, the reliance on intermediaries increases operational costs and inefficiencies in copyright enforcement. Dispute resolution remains a major issue, often favouring large corporations over individual creators and prolonging legal battles. Furthermore, centralized platforms restrict open access, limiting participation and inclusivity for global content creators.

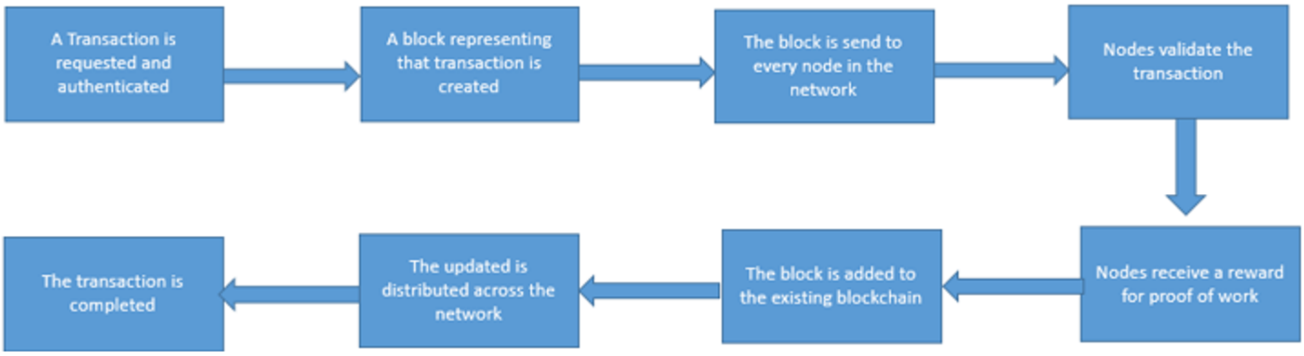

Figure 5. The blockchain workflow

Table 1. Literature review for business process model for open access CMS using permission less blockchain

Ref.	Title	Key Research Findings	Research Gaps
Weber et al. [20]	Untrusted Business Process Monitoring and Execution Using Blockchain	Explored blockchain for process monitoring and execution in untrusted environments.	Performance optimization and legal recognition challenges exist.
Mendling et al. [21]	Blockchains for Business Process Management—Challenges and Opportunities	Analyzed blockchain's impact on business process management.	Scalability and privacy integration issues persist.
Falazi et al. [22]	Modeling and Execution of Blockchain-Aware Business Processes	Introduced a modeling extension for business processes interacting with blockchain.	Standardization of modeling languages to fully support blockchain features is needed.
Guerreiro et al. [23]	Decentralized Business Process Control Using Blockchain: An Experience Report from Two Applications	Applied blockchain for process control in supply chain and vehicle registration.	Need for evaluating different modeling languages in blockchain applications.
Milani et al. [24]	Modelling Blockchain-Based Business Processes: A Comparative Analysis of BPMN vs CMMN	Compared BPMN and CMMN for blockchain-based process modeling.	Need for blockchain-specific modeling languages.
Stiehle and Weber [25]	Blockchain for Business Process Enactment: A Taxonomy and Systematic Literature Review	Developed a taxonomy for blockchain-based business process enactment.	Limited research on flexibility and scalability.
Jin and Ling [26]	A Blockchain-Based Copyright Management Scheme for Open-Source Software and Hardware Designs	Proposed a blockchain scheme for open-source copyright management.	Needs further validation in real-world scenarios.

To address these issues, a permission less blockchain-based CMS is proposed to ensure decentralization and transparency. By utilizing cryptographic hashing and smart contracts, the system enhances security, authenticity, and tamper-proof ownership verification. Automating copyright enforcement through smart contracts will reduce costs, eliminate intermediaries, and streamline royalty distribution. Additionally, blockchain-based consensus mechanisms will improve dispute resolution, making the process faster and fairer. To achieve scalability and efficiency, advanced blockchain solutions such as layer-2 protocols and sharing will be explored, ensuring global accessibility and seamless digital rights management.

The rest of the paper is organized as follows: Section 2 reviews the literature. Section 3 presents the proposed model and its architecture. Section 4 details the implementation using various optimization techniques. Section 5 analyses the results, followed by the conclusion and future improvements in the final section. Table 1 is the literature review for business process model for open access CMS using permission less blockchain.

2. LITERATURE REVIEW

Copyright management is essential in the creative industry to safeguard intellectual property rights and ensure fair compensation for creators. However, traditional copyright systems suffer from inefficiencies, lack of transparency, and susceptibility to infringement, making enforcement complex and costly [20, 21]. Blockchain technology offers a decentralized, immutable, and transparent approach to address these challenges, providing a more secure and efficient framework for digital rights management. This literature survey examines existing research on integrating blockchain into copyright management, emphasizing the development of business process models that leverage blockchain's capabilities to enhance security, automation, and scalability [22, 23].

The literature review highlights the evolution of CMSs, from traditional business process models to blockchain-based and hybrid approaches. Key findings indicate that traditional systems suffer from inefficiencies, reliance on intermediaries, and a lack of transparency. Business process models improve structure but remain limited in scalability and automation. Blockchain-based models enhance security, decentralization, and transparency, but they face challenges related to scalability, regulatory compliance, and interoperability. Hybrid models integrating blockchain with business process management (BPM) offer promising solutions but require optimization for high-speed transactions and real-world deployment. Future research should focus on developing standardized frameworks, AI-driven automation, and scalable blockchain solutions to address existing gaps [24-26].

3. PROPOSED MODEL

Integrating a Copyright Management System (CMS) with a business process model streamlines the protection and management of intellectual property throughout the organization. This integration ensures that copyright-related tasks are seamlessly incorporated into daily workflows, enhancing efficiency and reducing the risk of infringement. It promotes better collaboration among departments by providing a centralized platform for managing copyright information. Additionally, it aids in compliance with legal requirements and improves overall transparency in the handling of intellectual property. Ultimately, this integration fosters a more innovative and legally secure business environment [27-29]. Figure 6 presents a conceptual business model for a copyright protection system utilizing blockchain technology, arranged in a hexagonal layout.

Blockchain is a decentralized and distributed ledger technology that records transactions across multiple nodes, ensuring transparency, security, and immutability. Unlike traditional centralized systems, it operates without a central authority, using consensus mechanisms like Proof-of-Work (PoW) and Proof-of-Stake (PoS) for validation. Key features include decentralization, which eliminates intermediaries and reduces costs, and immutability, ensuring that recorded data cannot be altered or deleted [30-33]. Transparency allows publicly accessible records that enhance trust, while smart contracts enable automated licensing, royalty distribution, and content verification. Additionally, cryptographic security safeguards digital assets from tampering and unauthorized access.

Traditional copyright management relies on centralized authorities, leading to inefficiencies, disputes, and infringement risks. Blockchain, particularly permissionless blockchain, provides a decentralized solution by enabling authors to register their works with immutable proof of ownership [34-36]. It ensures transparent and verifiable transactions where every copyright transfer or licensing agreement is recorded and accessible to all participants. Smart contracts automate licensing, ensuring timely royalty payments, while tamper-proof recordkeeping prevents unauthorized modifications or deletions. With global accessibility, permissionless blockchain allows unrestricted participation, making it an ideal solution for open access content management [37].

3.1 A business process model for a CMS using blockchain

At the center is the core concept: "A business model for a copyright protection system using Blockchain," surrounded by various integral components. The Decentralized Copyright Registry serves as a central system for recording copyright ownership on the blockchain. Notification and Enforcement mechanisms ensure timely alerts and enforcement of copyright violations. Smart Contracts for Licensing automate and streamline the licensing process. Consulting and Legal Services provide professional support for the legal aspects of copyright management. Blockchain-based Monitoring and Enforcement utilizes blockchain to track and ensure compliance with copyright laws. Decentralized Royalty Management systems handle the management and distribution of royalties in a decentralized manner. Tokenization of Copyright Assets converts copyright assets into digital tokens for easier management and trading. Community Governance and Consensus Mechanisms involve community-driven governance to oversee the system. Education and Awareness Programs aim to educate stakeholders on the system's benefits and functionalities. Continuous Innovation and Collaboration encourage ongoing improvements and partnerships within the ecosystem. Interoperability and Integration ensure seamless interaction with other technologies and platforms. Finally, Tokenomics and Revenue Models provide economic structures to sustain and incentivize participation in the system. This comprehensive model outlines a holistic approach to modernizing copyright protection through blockchain technology. Figure 7 outlines a multi-layered architecture for integrating blockchain with business process management, consisting of several distinct layers.

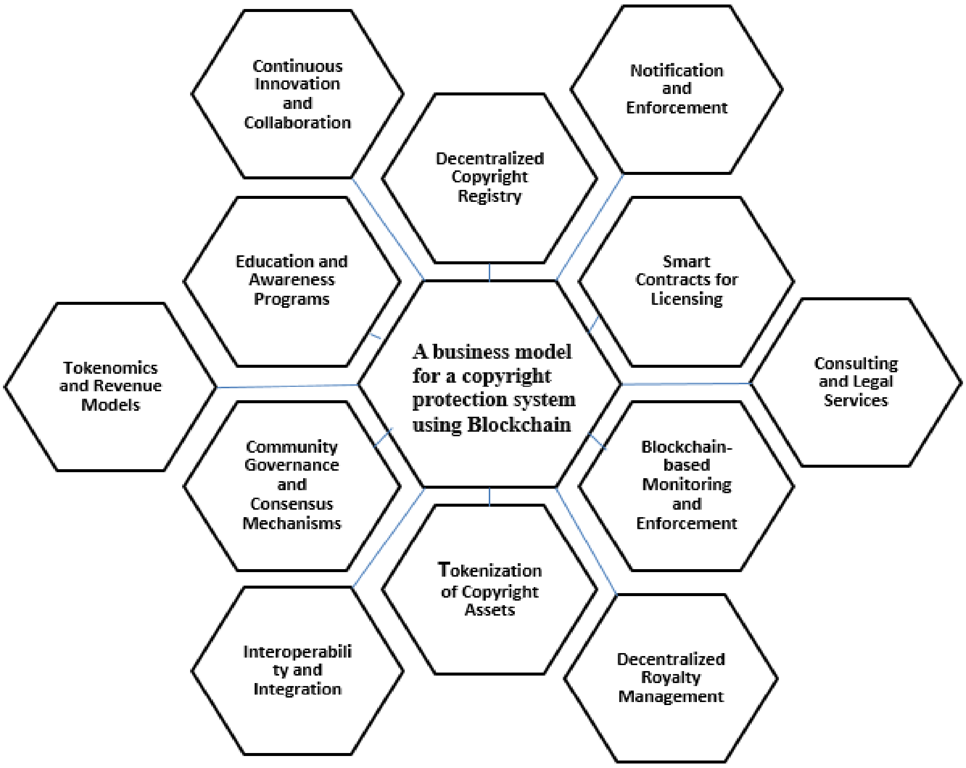

Figure 6. Overall model for CMS using blockchain technology

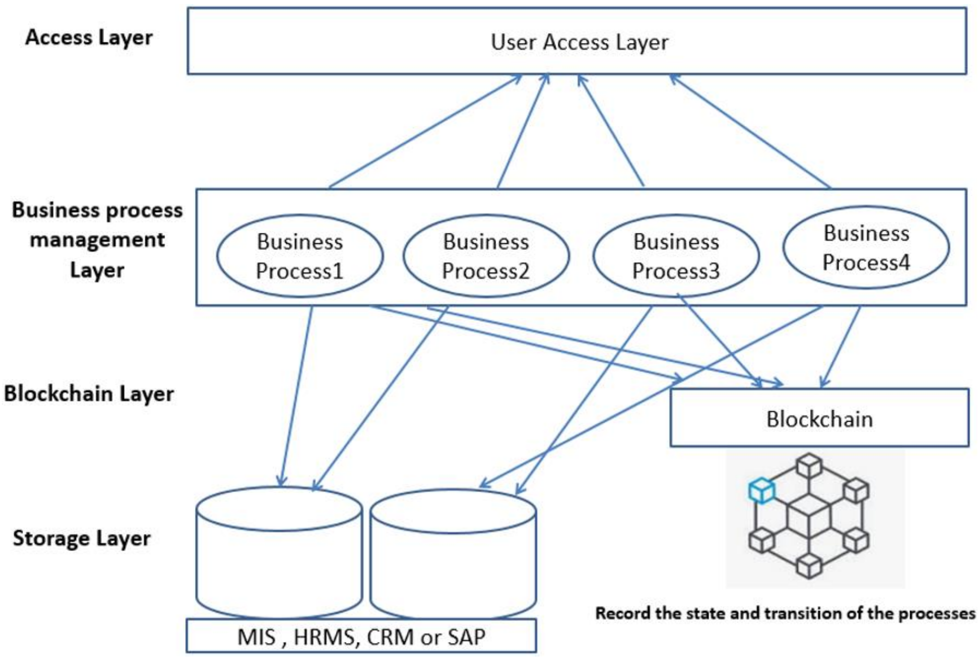

Figure 7. Block diagram of business model for CMS using blockchain technology

3.2 Architecture of business process model for a CMS using blockchain

The User Access Layer represents the interface through which users interact with the system, providing access to various business processes. The Business Process Management Layer contains multiple business processes (Business Process 1, Business Process 2, Business Process 3, and Business Process 4), each managed and coordinated to ensure seamless execution and interaction. The Blockchain Layer plays a crucial role in recording the state and transitions of the business processes, ensuring immutability, transparency, and providing a secure, decentralized method for tracking changes. The Storage Layer comprises traditional storage systems such as MIS (Management Information Systems), HRMS (Human Resource Management Systems), CRM (Customer Relationship Management), and SAP (Systems, Applications, and Products in Data Processing), storing data generated and used by the business processes. Interactions between these layers are well-defined: users access business processes directly via the User Access Layer; business processes interact with both the storage and blockchain layers; the blockchain layer securely records states and transitions; and the storage layer interfaces with business processes to store and retrieve necessary data. This architecture effectively integrates blockchain with business process management, enhancing the security, transparency, and efficiency of business operations, while leveraging traditional storage systems for comprehensive data management.

4. IMPLEMENTATION

Figure 8 proposes implementation model depicting the "Chain of Transformations" in a business process model for copyright protection system, outlining a series of progressive steps. It begins with the Business Process State Transition Model, which serves as the initial state transition framework for a business process. This model is then enhanced and optimized in the Optimized State Transition Model stage. This model is converted into a smart contract using the Solidity programming language, as described in the Convert the State Transition Model into Smart Contract step. Following optimization, the resulting smart contract undergoes further refinement in the Optimized Smart Contract stage. Finally, the optimized smart contract is transformed into an intelligent contract, still utilizing Solidity, in the Convert the Smart Contract into Intelligent Contract step. This flowchart illustrates the systematic progression from a conceptual model to a highly optimized intelligent contract through multiple stages of enhancement and coding.

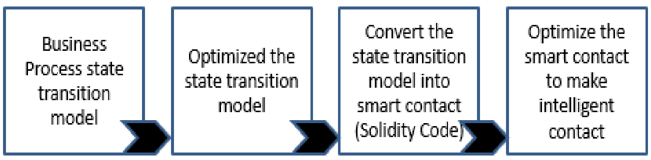

Figure 8. Chain of transformations for business process model for a CMS using blockchain

4.1 Business process state transition model for CPM

The business process state transition model for a copyright protection system involves a comprehensive flow, moving a copyright asset from creation to enforcement with distinct stages. Initially, the asset is created and registered, where the creator submits necessary details such as title, description, and ownership information. The asset then undergoes verification to ensure authenticity and originality, involving checks by a designated authority or automated system. Once verified, the asset is registered in the decentralized copyright registry, with its details entered into the blockchain for immutability and transparency. The registered asset becomes available for licensing through smart contracts, automating the licensing process and allowing potential licensees to acquire usage rights. Subsequently, the asset is monitored continuously for unauthorized use or infringement using blockchain-based tracking tools. In case of detected violations, enforcement actions are initiated, including notifications and legal actions against infringers, supported by immutable blockchain records. The system also manages and distributes royalties to the copyright owner by collecting licensing fees and distributing them automatically via smart contracts. If disputes arise, the dispute resolution stage utilizes blockchain's transparent records for fair and swift resolution. Finally, the asset's records are archived once it is no longer active or the copyright has expired, ensuring historical reference and compliance.

Summary of State Transitions:

1. Initial → Created
2. Created → Verified
3. Verified → Registered
4. Registered → Licensed
5. Licensed → Monitored
6. Monitored → Enforcement
7. Licensed → Royalty Management
8. Enforcement or Monitored → Dispute Resolution
9. Any Active State → Archived

4.2 Optimized state transition model using PASTRA algorithm

The optimized business process state transition model uses Processes States and Transition Reduction Algorithm (PSTRA) for a copyright protection system streamlines and refines each step for enhanced efficiency and clarity. The process begins with Creation and Submission, where the copyright asset is created and submitted for verification, including necessary details like title, description, and ownership information. The asset then moves to Verification and Registration, undergoing authenticity checks by a designated authority or automated system and being registered in the decentralized copyright registry for immutability and transparency. Once Verified and Registered, the asset becomes available for licensing through smart contracts in the Smart Contract Licensing stage, automating the licensing process and enabling potential licensees to acquire usage rights. The system ensures Monitoring and Enforcement by continuously tracking the asset for unauthorized use or infringement, utilizing blockchain-based tools to detect violations and initiating legal actions against infringers, supported by immutable blockchain records. Concurrently, Royalty Management handles the collection of licensing fees and the automatic distribution of royalties to the copyright owner through smart contracts. Should disputes arise, the Dispute Resolution stage leverages blockchain's transparent records to facilitate fair and swift resolution. Finally, the Archival stage ensures that once the asset is no longer active or the copyright has expired, its records are archived for historical reference and compliance.

Algorithm 1: Processes States and Transition Reduction Algorithm (PSTRA)

```
Input: State Transition Model
Begin:
1: Extract all states and transitions for the given diagram
2: Identify the bottlenecks in the states
        For each bottleneck
        Begin
        If (high volume of input data leading to slow processing)
            data Pre-processing → parallel processing →
```

allocate more resources to handle peak load
else if (long processing times due to complex algorithms)
algorithm Optimization → data Caching → batch Processing
else if (delays in data transfer due to large data volumes)
data Compression → efficient data structures → incremental transfer
else if (reprocessing and delays caused by inconsistent data)
data validation → error handling →
else if (multiple processes competing for the same resources)
resource scheduling → load balancing → prioritization
end
3: Analyze the process-flow to identify deviations and inefficiencies
For each deviation
Begin
If (deadlocks)
implement timeouts, deadlock detection, and regular state review
else if (unreachable states)
comprehensive testing, review transition logic, state coverage analysis
else if (redundant transitions)
simplify state diagram, optimize logic, review process design
else if (state Inconsistency & overlapping)
state separation, refactor states, process redesign
end
4: Evaluate each state and transition to extract non-value added activities.
5: Use Simplify Transition Logic (STL), Event-Driven Transitions (EDT) and
Asynchronous Processing (AP) tools to optimize the transitions.
6: Go to step 2 until (all states and transitions are not further optimized)
end
Output: Optimized State Transition Model

Summary of Optimized State Transitions:
1. Initial → Submitted
2. Submitted → Verified and Registered
3. Verified and Registered → Licensed
4. Licensed → Monitored and Enforced
5. Licensed → Royalty Managed
6. Monitored and Enforced → Dispute Resolved
7. Any Active State → Archived

To develop an optimized State Transition Model, we begin by extracting all states and transitions from the given diagram. This initial step is followed by identifying bottlenecks within the states. For each bottleneck, we implement specific solutions based on the nature of the issue. If the bottleneck is due to a high volume of input data leading to slow processing, we employ data pre-processing and parallel processing techniques, allocating more resources to handle peak loads. If long processing times are caused by complex algorithms, we focus on algorithm optimization, data caching, and batch processing. Delays in data transfer due to large data volumes are addressed through data compression, the use of efficient data structures, and incremental transfer methods. In cases of reprocessing and delays caused by inconsistent data, we enhance data validation and error handling. For bottlenecks where multiple processes compete for the same resources, we implement resource scheduling, load balancing, and prioritization. Next, we analyze the process flow to identify deviations and inefficiencies. For each deviation, we apply targeted solutions. Deadlocks are managed by implementing timeouts, deadlock detection, and regular state reviews. Unreachable states are addressed through comprehensive testing, reviewing transition logic, and conducting state coverage analysis. Redundant transitions are simplified by optimizing logic and reviewing the process design. State inconsistency and overlapping are resolved by separating states, refactoring, and redesigning processes. Each state and transition is evaluated to extract non-value-added activities. To further optimize transitions, we use tools such as Simplify Transition Logic (STL), Event-Driven Transitions (EDT), and Asynchronous Processing (AP). This cycle of analysis and optimization continues iteratively until no further optimizations can be achieved for all states and transitions. The result is an optimized State Transition Model. This approach involves the implementation of Algorithm 1, which is the Processes States and Transition Reduction Algorithm (PSTRA), for an Optimize**d** State Transition Model.

4.3 Convert the optimized state transition model into smart contract

Algorithm 2 Convert the state transition model into the Smart Contract

Input: Optimized State Transition Model
Begin
1: Define the Smart Contract
2: for each enumerated state
Begin
Setup state variable
Setup controls
end
3: for each transition
Begin
Create events
end
4: for each state
Begin
Create functions
end
5: Test the smart contact
end
Output: Smart contact

Figure 9 is to make sure the smart contract correctly reflects the logic and flow of the process state transition diagram; the process must be done in a number of methodical steps. The examples in Algorithm 1 First, look at the process state transition picture to see all the states that the system can be in and how it can change between them. Next, make a smart contract and use an Enum type in Solidity to list these states. In this step, a clear picture of each state is made in the contract. After that, set up the state variables and access controls that you will need to properly manage these states. To do this, you need to set variables for important pieces of information (like the contract owner, metadata, and a mapping for authorized rights) and use modifiers to control who can see what data (for example, only letting the contract owner see certain functions).

Once the basic setup is done, you can move on to making routines that handle the changes between states. As an example, a method that changes from the Created state to the Registered state should check for the right conditions and change the state variable as needed. Other functions would make sure that each change follows the logic shown in the diagram by giving and taking away licenses and transferring ownership. Implement events in the contract to keep track of important events and changes in the state. These events should be sent out when the state changes, creating a log that can be seen from outside the application. An event can be sent to record a change in a state, like when it goes from "Created" to "Registered." Lastly, the smart contract needs to be thoroughly tested to make sure it is being used correctly. Put the contract into action and run each function, making sure that state changes and events happen as planned. This part of testing makes sure that the smart contract accurately shows the process state transition diagram, with each state and transition being correctly implemented and managed. In Algorithm 2 explain Convert the state transition model into the Smart Contract. Figure 9 shows convert state transition model for CMS into smart contract

Listing 1.1: Sample code to convert State Transition Model for Copyright Management System into Smart Contact

```
pragma solidity ^0.8.0;
contract CopyrightManagementSystem {
  // Step 1: Define the contract and enumerate states
  enum State { Created, Registered, Licensed, Transferred }
  State public currentState;
  // Step 2: Set up state variables and access control
  address public owner;
  string public metadata;
  mapping(address => bool) public authorizedLicenses;
  // Step 3: Implement events
  event StateChanged(State oldState, State newState);
  event OwnershipTransferred(address indexed previousOwner, address indexed
newOwner);
  event LicenseGranted(address indexed licensee);
  event LicenseRevoked(address indexed licensee);

  // Modifier to check if the caller is the owner
  modifier onlyOwner() {
    require(msg.sender == owner, "Caller is not the owner");
  }
  // Constructor to initialize the contract
  constructor(string memory _metadata) {
    owner = msg.sender;
    metadata = _metadata;
    currentState = State.Created;
    emit StateChanged(State.Created, currentState);
  }
  // Step 4: Create state transition functions
  // Function to register the copyright
  function register() public onlyOwner {
    require(currentState == State.Created, "Invalid state transition");
    currentState = State.Registered;
    emit StateChanged(State.Created, currentState);
  }
  // Function to grant a license
  function grantLicense(address _licensee) public onlyOwner {
    require(currentState == State.Registered, "Cannot grant license in current state");
    authorizedLicenses[_licensee] = true;
    emit LicenseGranted(_licensee);
  }
  // Function to revoke a license
  function revokeLicense(address _licensee) public onlyOwner {
    require(authorizedLicenses[_licensee], "Licensee is not authorized");
    authorizedLicenses[_licensee] = false;
    emit LicenseRevoked(_licensee);
  }
  // Function to transfer ownership
  function transferOwnership(address newOwner) public onlyOwner {
    require(currentState == State.Registered || currentState == State.Licensed, "Invalid
state for transfer");
    emit OwnershipTransferred(owner, newOwner);
    owner = newOwner;
    currentState = State.Transferred;
    emit StateChanged(State.Registered, currentState);
  }
  // Step 5: Test the smart contract (tests would typically be written in a separate test
file)
  // Example:
  // - Deploy the contract
  // - Register the copyright
  // - Grant and revoke licenses
  // - Transfer ownership
  // - Check event logs and state variables
}
```

Figure 9. Convert state transition model for CMS into smart contract

```
Listing 1.2: Sample code to convert smart contact into intelligent contact

pragma solidity ^0.8.0;
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
contract CopyrightProtection is Ownable, ERC721 {
    enum State { Created, Registered, Verified, InfringementDetected, Resolved }
    State public currentState;
    address public verifier,authority;
    modifier onlyVerifier() {
        require(msg.sender == verifier, "Not authorized");
        _; }
    modifier onlyAuthority() {
        require(msg.sender == authority, "Not authorized");
        _;  }
    modifier inState(State _state) {
        require(currentState == _state, "Invalid state transition");
        _;     }
    event StateChanged(State oldState, State newState);
    constructor(address _verifier, address _authority) ERC721("CopyrightAsset", "CPA")
      { verifier = _verifier;
        authority = _authority;
        currentState = State.Created;  }
    function registerAsset() external onlyOwner inState(State.Created) {
        currentState = State.Registered;
        _mint(msg.sender, 1);  // Mint an NFT representing the asset
        emit StateChanged(State.Created, currentState); }
    function verifyAsset() external onlyVerifier inState(State.Registered) {
        currentState = State.Verified;
        emit StateChanged(State.Registered, currentState);     }
    function reportInfringement() external onlyAuthority inState(State.Verified) {
        currentState = State.InfringementDetected;
        emit StateChanged(State.Verified, currentState); }
    function resolveInfringement() external onlyAuthority inState(State.InfringementDetected)
        { currentState = State.Resolved;
        emit StateChanged(State.InfringementDetected, currentState);
    } }
```

Figure 10. Convert smart contract into intelligent contract

4.4 Optimize the smart contract to make intelligent contract

Figure 10 is to create an optimized smart contract; we start by opening the original contract. For each function within the contract, we generate a new intermediary contract. We then analyze each line within the function. If the data type is Uint256, we check if the value is less than 128 and, if so, change it to Uint128. If an array is used, we assess whether a mapping would be more appropriate and make the change if needed. Variables are converted to fixed-size where possible. Additionally, we replace division or modulo operators with addition, subtraction, or multiplication to optimize performance.

Next, we examine each loop within the function. If a loop is unbounded, we avoid using it. Any dead code within the loop body is eliminated. For repeated storage calls, we store the value in a local variable instead of repeatedly accessing storage. Constant expressions that always evaluate to true are removed. Repeated computations are replaced with a constant value computed once before the loop starts. Algorithm 3 optimizes the functionality of Smart Contracts, resulting in the creation of Intelligent Contacts that are optimized. Figure 10 presents a code snippet that demonstrates the process of transforming a smart contract into an intelligent contract. After optimizing each line and loop in the function, we combine all the new intermediary contracts into a single new contract, referred to as the optimized contract. Finally, we ensure that the optimized contract is behaviorally equivalent to the original contract using a bisimulation model. The result is an optimized smart contract that maintains the same functionality as the original while being more efficient and streamlined.

The efficiency of smart contracts, which are self-executing contracts with the terms of the agreement directly written into code, is evaluated through several key metrics. Gas consumption measures the amount of computational effort required to execute operations on a blockchain network, crucial for cost estimation and optimization. Execution time refers to the duration taken for a smart contract to complete its operations, influencing the responsiveness and scalability of applications. Storage efficiency assesses how effectively data is managed and stored within the blockchain, impacting overall performance and cost-effectiveness. Security audits ensure that smart contracts are robust against vulnerabilities and attacks, safeguarding digital assets and transactions. Function call efficiency evaluates how efficiently functions within smart contracts are executed, optimizing resource allocation and enhancing operational speed. Together, these metrics provide a comprehensive assessment of smart contract efficiency, guiding developers and stakeholders in maximizing performance and reliability in blockchain applications.

4.5 Limitations of the proposed model

While the proposed model effectively optimizes business process transitions for copyright management, several limitations exist. First, the complexity of the multi-stage transformation from state transition models to intelligent contracts may introduce challenges in implementation and debugging. Second, despite optimization efforts, blockchain gas fees and storage costs remain a concern, particularly for large-scale deployments. Third, the approach heavily relies on Solidity, limiting its applicability to other blockchain platforms with different smart contract languages. Fourth, security vulnerabilities may still exist despite optimizations, necessitating continuous auditing and updates. Lastly, real-world adoption may be hindered by legal and regulatory challenges, requiring further standardization and compliance measures.

Algorithm 3 Optimize the Smart Contact to make Intelligent Contact

```
Input: Smart Contact
Begin
1:      Open the Contract
2:      For each function
3:      Begin
4:      Create new intermediary contact
5:      For each line
6:      Begin if (type is Uint256)
        check the value less than 128 change to Uint128
        else if (array)
        check the requirement change to mapping
        else if (variable): change to fixed size
        else if (operators are division or modulo)
        change to addition, subtraction or multiplication.
        end
7:      For each loop
        Begin   if (unbounded)
                Avoid that loop
        if (dead code)
                eliminate the code with in the loop body
        else If (repeated storage call)
        Store the value in local variable instead of storage
        else if (constant expressions)
        eliminate expression if that condition always evaluate
        to true
        else if (repeatedly computations)
                replaced by a constant value computed once before starting of the loop
        end
8:      end
9:      Combine the all-new intermediary contacts into a single new contact called optimized contract
10:     Check the optimized contact behaviorally equivalent to the original contact using bisimulation model
end
Output: Optimized Smart Contract
```

5. COMPREHENSIVE COMPARISON AND ANALYSIS

The comparative analysis of different models integrating blockchain with copyright management and business process management highlights the unique strengths and limitations of each approach. This table provides an overview of various works related to the integration of copyright management with blockchain technology, assessed based on several key criteria: decentralization, transparency and traceability, cost reduction, rights management and protection, enhanced security, cooperation, coordination and communication, optimization, and efficiency and automation. In previous study [5], authors focus on copyright using watermarking and cryptography, offering transparency, traceability, rights management, and efficiency, but lacking decentralization, cost reduction, enhanced security, cooperation, coordination, communication, and optimization. In contrast, the study [7] explores blockchain for Business Process Models (BPM), achieving decentralization, transparency, traceability, rights management, enhanced security, cooperation, coordination, communication, and efficiency, but not cost reduction or optimization. Similarly, article [9] use blockchain for CMS, achieving decentralization, transparency, traceability, rights management, and efficiency, but not cost reduction, enhanced security, cooperation, coordination, communication, or optimization. The proposed model integrates copyright management and BPM with blockchain, addressing all the evaluated criteria. It achieves decentralization, transparency, traceability, cost reduction, rights management, enhanced security, cooperation, coordination, communication, optimization, and efficiency and automation. This comprehensive comparison highlights the strengths and weaknesses of each approach and underscores the unique benefits of the proposed model, showcasing its superiority in optimizing and enhancing various facets of copyright management through blockchain integration. Table 2 describes comparison of various existing approaches and proposed approach with various parameters. Each parameter is scored on a scale of 0 to 5: 0: Not addressed 1-2: Partially addressed 3-4: Well addressed 5: Fully addressed.

To visualize the comparison, the scores for each parameter will be plotted using a bar chart.

In Figure 11, the Proposed Model stands out as a comprehensive solution, addressing all critical parameters required for blockchain-enabled copyright management and business process modelling (BPM). Unlike earlier works and intermediate models, which excel only in specific areas, the Proposed Model achieves balanced and complete coverage of decentralization, transparency, cost reduction, rights management, enhanced features, optimization, and efficiency. This holistic approach ensures a robust system that meets the multifaceted demands of modern copyright management and BPM.

Decentralization is a key strength of the Proposed Model. While early works [5, 6] relied on centralized mechanisms prone to single points of failure and scalability issues, intermediate models such as works [7, 8] partially leveraged blockchain's decentralization but retained hybrid dependencies. In contrast, the Proposed Model eliminates central authority reliance, enhancing system resilience and trustworthiness across stakeholders.

Cost reduction is another area where the Proposed Model excels. Earlier and intermediate works failed to prioritize cost efficiency, relying on intermediaries that increased operational expenses. By integrating blockchain and BPM, the Proposed Model automates processes, reduces intermediary dependency, and streamlines operations, leading to significant cost savings for users and organizations.

Table 2. Comparison of existing and proposed approaches with various parameters

Ref.	Decentralization	Transparency & Traceability	Cost Reduction	Rights Management & Protection	Enhanced Features	Optimization	Efficiency & Automation	Total Score
Meng et al. [5]	0	4	0	4	2	0	3	13
Piva et al. [6]	0	4	0	4	2	0	3	13
Garcia-Garcia et al. [7]	4	4	0	2	4	2	4	20
Di Ciccio et al. [8]	4	4	0	2	4	2	4	20
Jing et al. [9]	4	4	0	2	4	2	4	20
Ding et al. [11]	4	4	0	2	4	2	4	20
Propose Model	5	5	5	5	5	5	5	35

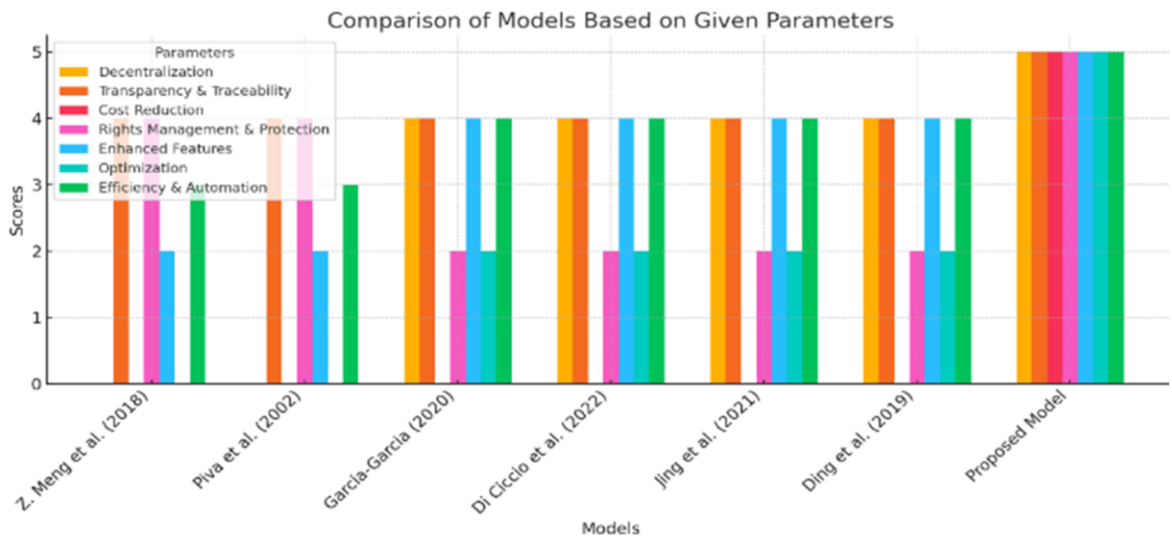

Figure 11. Comprehensive comparison and analysis

Automation and optimization are fully realized in the Proposed Model, in contrast to earlier models that relied on cryptographic techniques and standalone processes or intermediate models that offered partial automation. Leveraging smart contracts and advanced integrations, the Proposed Model automates and optimizes operations comprehensively, improving efficiency while reducing manual intervention and errors.

The Proposed Model also enhances security, collaboration, coordination, and communication—features often absent or only partially addressed in earlier works. By combining blockchain's inherent capabilities with BPM integration, it fosters a secure and collaborative environment, ensuring seamless coordination among stakeholders.

The evolution of blockchain adoption is evident in the progression from early works to the Proposed Model. Early research focused on cryptographic watermarking and transparency, lacking decentralization, cost efficiency, and optimization. Intermediate models advanced blockchain use for decentralization and transparency but fell short in realizing its full potential. The Proposed Model represents the culmination of these efforts, integrating all parameters to create a robust, efficient, and fully optimized solution.

6. CONCLUSION AND FUTURE RESEARCH

This paper presents the integration of a Business Process Model for a CMS using blockchain, which holds significant promise for enhancing various facets of business operations, such as security, transparency, traceability, and efficiency. While surveys indicate substantial opportunities for improving business processes through this integration, they also highlight concerns around trust variability and transaction uncertainties. The proposed model involves a chain of transformations, starting with converting a state transition model into an optimized transition model using the PSTRA algorithm. This optimized transition model is then converted into smart contracts using a conversion algorithm. Finally, these smart contracts are translated into intelligent contracts using an optimized algorithm. This pioneering approach aims to execute cooperative and collaborative processes on blockchain, albeit with acknowledged challenges in optimizing for cost and performance. When compared to other models, the proposed optimized model demonstrates superior results in performance parameters such as decentralization, transparency, traceability, cost reduction, rights management and protection, and enhanced security, cooperation, coordination, communication, and efficiency. The approach presented in this article seeks to exploit the synergies between Copyright Management and Business Process Management (BPM), providing integrated solutions that cover cost effectiveness, heightened security, transference, and improved operations.

Future research can focus on addressing these limitations through several key directions. First, developing automated tools for seamless transformation from state transition models to intelligent contracts can reduce complexity and debugging challenges. Second, exploring layer-2 scaling solutions or alternative consensus mechanisms can help mitigate blockchain gas fees and storage costs, enhancing efficiency for large-scale deployments. Third, extending the model to support multiple blockchain platforms by integrating cross-chain interoperability frameworks and smart contract languages beyond Solidity can improve adaptability. Fourth, enhancing security by integrating formal verification techniques and AI-driven vulnerability detection can strengthen smart contract robustness. Lastly, studying legal and regulatory frameworks to develop standardized compliance protocols for blockchain-based copyright protection systems can facilitate broader real-world adoption.

REFERENCES

[1] Nelaturu, K., Beillahi, S.M., Long, F., Veneris, A. (2021). Smart contracts refinement for gas optimization. In 2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Paris, France, pp. 229-236. https://doi.org/10.1109/BRAINS52497.2021.9569819

[2] Hu, W., Fan, Z., Gao, Y. (2019). Research on smart contract optimization method on blockchain. IT Professional, 21(5): 33-38. https://doi.org/10.1109/MITP.2019.2923604

[3] Renieri, M. (2020). Ethereum smart contracts optimization. Diss. University of Camerino.

[4] Mir, N., Khan, M.A. (2020). Copyright protection for online text information: Using watermarking and cryptography. In 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, pp. 1-4. https://doi.org/10.1109/ICCAIS48893.2020.9096817

[5] Meng, Z., Morizumi, T., Miyata, S., & Kinoshita, H. (2018, July). Design scheme of copyright management system based on digital watermarking and blockchain. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, pp. 359-364. https://doi.org/10.1109/COMPSAC.2018.10258

[6] Piva, A., Bartolini, F., Barni, M. (2002). Managing copyright in open networks. IEEE Internet Computing, 6(3): 18-26. https://doi.org/10.1109/MIC.2002.1003126

[7] Garcia-Garcia, J.A., Sánchez-Gómez, N., Lizcano, D., Escalona, M.J., Wojdyński, T. (2020). Using blockchain to improve collaborative business process management: Systematic literature review. IEEE Access, 8: 142312-142336. https://doi.org/10.1109/ACCESS.2020.3013911

[8] Di Ciccio, C., Meroni, G., Plebani, P. (2022). On the adoption of blockchain for business process monitoring. Software and Systems Modeling, 21(3): 915-937. https://doi.org/10.1007/s10270-021-00959-x

[9] Jing, N., Liu, Q., Sugumaran, V. (2021). A blockchain-based code copyright management system. Information Processing & Management, 58(3): 102518. https://doi.org/10.1016/j.ipm.2021.102518

[10] Wang, P., Li, Y., Li, F., Dong, X.H., Chen, P. (2019). Secure and traceable copyright management system based on blockchain. In 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, pp. 1243-1247. https://doi.org/10.1109/ICCC47050.2019.9064101

[11] Ding, Y., Yang, L., Shi, W., Duan, X. (2019). The digital copyright management system based on blockchain. In 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, pp. 63-68.

https://doi.org/10.1109/CCET48361.2019.8989370

[12] Köpke, J., Meroni, G., Salnitri, M. (2023). Designing secure business processes for blockchains with SecBPMN2BC. Future Generation Computer Systems, 141: 382-398. https://doi.org/10.1016/j.future.2022.11.013

[13] García-Bañuelos, L., Ponomarev, A., Dumas, M., Weber, I. (2017). Optimized execution of business processes on blockchain. In Business Process Management: 15th International Conference (BPM 2017) Barcelona, Spain, pp. 130-146. https://doi.org/10.1007/978-3-319-65000-5_8

[14] Viriyasitavat, W., Da Xu, L., Niyato, D., Bi, Z., Hoonsopon, D. (2022). Applications of blockchain in business processes: A comprehensive review. IEEE Access, 10, 118900-118925. https://doi.org/10.1109/ACCESS.2022.3217794

[15] Mendling, J., Weber, I., Aalst, W.V.D., et al. (2018). Blockchains for business process management-challenges and opportunities. ACM Transactions on Management Information Systems (TMIS), 9(1): 1-16. https://doi.org/10.1145/3183367

[16] Han, P., Sui, A., Jiang, T., Gu, C. (2020). Copyright certificate storage and trading system based on blockchain. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, pp. 611-615. https://doi.org/10.1109/AEECA49918.2020.9213631

[17] Nizamuddin, N., Salah, K., Azad, M.A., Arshad, J., Rehman, M.H. (2019). Decentralized document version control using Ethereum blockchain and IPFS. Computers & Electrical Engineering, 76, 183-197. https://doi.org/10.1016/j.compeleceng.2019.03.014

[18] Yang, F., Shi, Y., Wu, Q., Li, F., Zhou, W., Hu, Z., Xiong, N., Zhang, Y. (2019). The survey on intellectual property based on blockchain technology. In 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), pp. 743-748. https://doi.org/10.1109/ICPHYS.2019.8780125

[19] López-Pintado, O., Dumas, M., García-Bañuelos, L., Weber, I. (2019). Interpreted execution of business process models on blockchain. In 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC), Paris, France, pp. 206-215. https://doi.org/10.1109/EDOC.2019.00033

[20] Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J. (2016). Untrusted business process monitoring and execution using blockchain. In Business Process Management: 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, pp. 329-347. https://doi.org/10.1007/978-3-319-45348-419

[21] Mendling, J., Weber, I., Aalst, W.V.D., Brocke, J.V., et al. (2018). Blockchains for business process management-challenges and opportunities. ACM Transactions on Management Information Systems (TMIS), 9(1): 1-16. https://doi.org/10.1145/3183367

[22] Falazi, G., Hahn, M., Breitenbücher, U., Leymann, F. (2019). Modeling and execution of blockchain-aware business processes. SICS Software-Intensive Cyber-Physical Systems, 34: 105-116. https://doi.org/10.1007/s00450-019-00399-5

[23] Guerreiro, S.L., Silva, D., Rosado, T., Vasconcelos, A., Correia, M., Sousa, P. (2020). Decentralized business process control using blockchain. International Journal of Conceptual Modeling, 15(15): 1-41. https://doi.org/10.18417/emisa.15.15

[24] Milani, F., Garcia-Banuelos, L., Filipova, S., Markovska, M. (2021). Modelling blockchain-based business processes: A comparative analysis of BPMN vs CMMN. Business Process Management Journal, 27(2): 638-657. https://doi.org/10.1108/bpmj-06-2020-0263

[25] Stiehle, F., Weber, I. (2022). Blockchain for business process enactment: A taxonomy and systematic literature review. In International Conference on Business Process Management, Münster, Germany, pp. 5-20. https://doi.org/10.1007/978-3-031-16168-1_1

[26] Jin, Y., Ling, L. (2024). A blockchain-based copyright management scheme for open-source software and hardware designs. Tsinghua Science and Technology, 29(1): 121-135. https://doi.org/10.26599/TST.2023.9010112

[27] Murala, D.K. (2024). METAEDUCATION: State-of-the-art methodology for empowering feature education. IEEE Access, 12: 57992-58020. https://doi.org/10.1109/ACCESS.2024.3391903

[28] Murala, D.K., Panda, S.K., Dash, S.P. (2023). MedMetaverse: Medical care of chronic disease patients and managing data using artificial intelligence, blockchain, and wearable devices state-of-the-art methodology. IEEE Access, 11: 138954-138985. https://doi.org/10.1109/ACCESS.2023.3340791

[29] Murala, D.K., Panda, S.K., Sahoo, S.K. (2023). Securing electronic health record system in cloud environment using blockchain technology. In: Panda, S.K., Mishra, V., Dash, S.P., Pani, A.K. (eds) Recent Advances in Blockchain Technology. Intelligent Systems Reference Library, vol 237. Springer, Cham. https://doi.org/10.1007/978-3-031-22835-3_4

[30] Murala, D.K., Panda, S.K. (2023). Artificial intelligence in the development of metaverse. Metaverse and Immersive Technologies: An Introduction to Industrial, Business and Social Applications, 407-436. https://doi.org/10.1002/9781394177165.ch15

[31] Murala, D.K., Panda, S.K. (2023). Metaverse: A study on immersive technologies. Metaverse and Immersive Technologies: An Introduction to Industrial, Business and Social Applications, 1-41. https://doi.org/10.1002/9781394177165.ch1

[32] Murala, D.K., Panda, S.K. (2023). The internet of things in developing Metaverse. Metaverse and Immersive Technologies: An Introduction to Industrial, Business and Social Applications, 437-465. https://doi.org/10.1002/9781394177165.ch16

[33] Chandrashekhar, A., Saheb, S.H., Panda, S.K., Balamurugan, S., Peng, S.L. (2023). Metaverse and Immersive Technologies: An Introduction to Industrial, Business and Social Applications. John Wiley & Sons.

[34] Owusu-Antwi, K., Amenuvor, F.E. (2023). Understanding the metaverse: A review of virtual worlds and augmented reality environments. Current Journal of Applied Science and Technology, 42(23): 42-48. https://doi.org/10.9734/cjast/2023/v42i234172

[35] Rao, K.V., Murala, D.K., Panda, S.K. (2023). Blockchain: A study of new business model. In: Panda, S.K., Mishra, V., Dash, S.P., Pani, A.K. (eds) Recent Advances in Blockchain Technology. Intelligent Systems Reference Library, vol 237. Springer, Cham. https://doi.org/10.1007/978-3-031-22835-3_9

[36] Varaprasada Rao, K., Panda, S.K. (2023). A Design Model of Copyright Protection System Based on Distributed Ledger Technology. In: Satapathy, S.C., Lin, J.CW., Wee, L.K., Bhateja, V., Rajesh, T.M. (eds) Computer Communication, Networking and IoT. Lecture Notes in Networks and Systems, vol 459. Springer, Singapore. https://doi.org/10.1007/978-981-19-1976-3_17

[37] Varaprasada Rao, K., Panda, S.K. (2023). Secure electronic voting (E-voting) system based on blockchain on various platforms. In: Satapathy, S.C., Lin, J.CW., Wee, L.K., Bhateja, V., Rajesh, T.M. (eds) Computer Communication, Networking and IoT. Lecture Notes in Networks and Systems, vol 459. Springer, Singapore. https://doi.org/10.1007/978-981-19-1976-3_18